



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formatos para prácticas de laboratorio

| CARRERA | PLAN DE ESTUDIO | CLAVE ASIGNATURA | NOMBRE DE LA ASIGNATURA |
|------------------------|-----------------|------------------|-------------------------------------|
| Ing. En Comp. y L.S.C. | 2003-1 | 5038 | Programación Orientada a Objetos II |

| PRÁCTICA No. | LABORATORIO DE | Ingeniero en Computación y Licenciado en Sistemas Computacionales | DURACIÓN (HORA) |
|--------------|-----------------------|---|-----------------|
| 2 | NOMBRE DE LA PRÁCTICA | Escritura URL | 2 |

1. INTRODUCCIÓN

Muchas páginas HTML contienen formas que permiten introducir datos en el servidor. Después de teclear la información requerida e iniciar la petición pulsando un botón, el navegador escribe los datos en la URL a través de la red. Después la otra parte de la conexión los procesa y envía de vuelta una respuesta.

2. OBJETIVO (COMPETENCIA)

El alumno conocerá y practicará el procedimiento general de escritura URL.

3. FUNDAMENTO

En la comunicación con URL en general se debe enviar primero información de consulta antes de recuperar los datos de la URL. Los pasos son los siguientes:

1. Establecer un objeto URLConnection, a partir de la URL.
2. Establecer los parámetros de la conexión deseada
3. Conectarse con el URL destino
4. Escribir los datos de consulta de la conexión
5. Leer la respuesta.

El método **openConnection** crea y devuelve un objeto URLConnection para lectura y escritura o ambas.

| | | | |
|--|---|--------------------|--|
| Formuló Ing. Alicia del R. López Aguirre | Revisó M.C. Gloria Etelbina Chávez Valenzuela y LSC Monica Lam Mora | Aprobó | Autorizó M.C. Miguel Ángel Martínez Romero |
| Maestro | Coordinador de Programa Educativo | Gestión de Calidad | Director de la Facultad |



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formatos para prácticas de laboratorio

```
URLConnection connection = urlobj.openConnection();
```

Para activar la escritura utilizamos:

```
connection.setdoOutput(true);
connection.connect();
```

:

Ahora se puede escribir al URL, utilizando OutputStream de la conexión o leer con InputSream.

//escritura

```
OutputStream out= connection.getOutputStream();
```

// lectura

```
InputStream in= connection.getInputStream();
```

Al escribir en una conexión URLConnection se envían datos a un servidor mediante una consulta Post. Una consulta

Post incluye:

- Un URL que especifica un programa cgi.
- Encabezados Content-Type y Content-Length
- Un cuerpo del mensaje

Nota: Nos debemos preocupar solo del cuerpo del mensaje.(convención form de html). Clase URLEncoder y URLDecoder (UTF-8).

Ejemplo 1:

El siguiente código muestra como se puede realizar un procedimiento de escritura a una URL.

```
import java.io.*;
import java.net.*;
public class ReverseTest
{
public static void main(String[] args)
{
try {
if (args.length != 1)
{
System.err.println("Usage: java ReverseTest string_to_reverse");
System.exit(1);
}
}
}
```



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD

Formatos para prácticas de laboratorio

```

    }
    String stringToReverse = URLEncoder.encode(args[0]);
    URL url = new URL("http://java.sun.com/cgi-bin/backwards");
    URLConnection connection = url.openConnection();
    connection.setDoOutput(true);
    PrintWriter writer = new PrintWriter(connection.getOutputStream());
    writer.println("string=" + stringToReverse);
    writer.close();
    BufferedReader reader = new BufferedReader(new InputStreamReader(connection.getInputStream()));
    String inputLine;
    while ((inputLine = reader.readLine()) != null)
    {
        System.out.println(inputLine);
    }
    reader.close();
} catch (MalformedURLException me) {System.err.println("MalformedURLException: " + me);}
catch (IOException ioe) { System.err.println("IOException: " + ioe); }
}
}

```

Explicación del código anterior

Examinemos el programa y veamos como trabaja. Primero, el programa procesa los argumentos de la línea de comandos:

```

if (args.length != 1)
{
    System.err.println("Usage: java ReverseTest string_to_reverse");
    System.exit(1);
}

```

```
String stringToReverse = URLEncoder.encode(args[0]);
```

Las líneas anteriores aseguran que el usuario proporciona uno y sólo un argumento de la línea de comandos del programa y lo codifica. El argumento de la línea de comandos es la cadena a invertir por el script **cgi-bin backwards**. El argumento de la línea de comandos podría tener espacios u otros caracteres no alfanuméricos. Estos caracteres deben ser codificados porque podrían suceder varios procesos en la cadena en el lado del servidor. Esto se consigue mediante la clase **URLEncoder**.

Luego el programa crea el objeto **URL** -- la URL para el script **backwards** en **java.sun.com**. Como lo muestra la siguiente línea.

```
URL url = new URL("http://java.sun.com/cgi-bin/backwards");
```



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD

Formatos para prácticas de laboratorio

El programa crea una URLConnection y abre un stream de salida sobre esa conexión. El stream de salida está filtrado a través de un PrintStream.

```
URLConnection connection = url.openConnection();
PrintStream outStream = new PrintStream(connection.getOutputStream());
```

La segunda línea anterior llama al método **getOutputStream()** sobre la conexión. Si URL no soporta salida, este método lanza una UnknownServiceException. Si la URL soporta salida, este método devuelve un stream de salida que está conectado al stream de entrada estandar de la URL en el lado del servidor -- la salida del cliente es la entrada del servidor.

Luego, el programa escribe la información requerida al stream de salida y cierra el stream.

```
outStream.println("string=" + stringToReverse);
outStream.close();
```

Esta línea escribe en el canal de salida utilizando el método **println()**. Como puedes ver, escribir datos a una URL es tan sencillo como escribir datos en un stream. Los datos escritos en el stream de salida en el lado del cliente son la entrada para el script **backwards** en el lado del servidor. El programa ReverseTest construye la entrada en la forma requerida por el script mediante la concatenación **string=** para codificar la cadena.

Frecuentemente, como en este ejemplo, cuando escribe en una URL está pasando información al script cgi-bin que lee la información que usted escribe, realiza alguna acción y luego envía la información de vuelta mediante la misma URL. Por lo que querrás leer desde la URL después de haber escrito en ella. El programa ReverseTest los hace de esta forma:

```
DataInputStream inStream = new
DataInputStream(connection.getInputStream());
String inputLine;

while (null != (inputLine = inStream.readLine()))
{
    System.out.println(inputLine);
}
inStream.close();
```

Nota: Cuando ejecutes el programa ReverseTest utilizando **Invierteme** como argumento, deberás ver esta salida:

```
Invierteme
reversed is:
emetreivnl
```



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formatos para prácticas de laboratorio

Este ejemplo fué tomado de la siguiente fuente
<http://ji.ehu.es/LMAlonso/SW/java/Bib/tutorjava/html/red/urlsreadingwriting.html>

4. PROCEDIMIENTO (DESCRIPCIÓN)

| A) | EQUIPO NECESARIO | MATERIAL DE APOYO |
|----|------------------|-------------------|
|----|------------------|-------------------|

Equipo de computo con:

SDK de Java

Editor de Java – NetBeans

Práctica impresa

1.-Escriba un programa que se conecta a la siguiente url

<http://yaqui.mxl.uabc.mx/~lopez/script.php>

Esta URL te conectará aun script escrito en php llamado precisamente script.php, después de realizar la conexión, tu programa deberá escribir mediante la URL los siguientes datos al script

nombre="Mario Molina"

email="Mario_Molina@yahoo.com"

Finalmente deberás leer los datos desde la URL y los mostrarás en la consola.

| C) | CÁLCULOS Y REPORTE |
|----|--------------------|
|----|--------------------|

Se realizarán preguntas al alumno para verificar la comprensión del tema y se deberá entregar un reporte impreso del código elaborado para el ejercicio 1. Colocar comentarios personales a dicho código.

5. RESULTADOS Y CONCLUSIONES

Al finalizar la práctica el alumno será capaz de realizar programas que impliquen la comunicación con URL.

6. ANEXOS

www.java.sun.com

7. REFERENCIAS

<http://ji.ehu.es/LMAlonso/SW/java/Bib/tutorjava/html/red/urlsreadingwriting.html>